

УДК 004.4:004.8:338.48

Научная статья

 <https://doi.org/10.35330/1991-6639-2026-28-4-46-61>

 LSXPHG

Проектирование iOS-приложения интеллектуального туристического гида с интеграцией AI-сервисов: архитектура, UX и технологии

К. А. Бережной

Российский экономический университет имени Г. В. Плеханова
115054, Россия, Москва, Стремянный переулок, 36

Аннотация. Мобильные приложения стали основным каналом сопровождения путешественника на всех этапах поездки, а развитие искусственного интеллекта формирует у пользователей ожидание персонализированного «умного гида». При этом существующие туристические сервисы редко сочетают глубокую персонализацию, контекстную маршрутизацию, защиту приватности данных и устойчивую работу при нестабильном интернет-соединении.

Цель исследования – разработать и обосновать подход к проектированию iOS-приложения интеллектуального туристического гида с интеграцией AI-сервисов, включая выбор архитектуры, технологического стека, методов персонализации и UX-решений.

Материалы и методы исследования. Выполнен анализ 47 приложений категории «Travel» в App Store и 17 специализированных AI-сервисов; проведен сравнительный анализ подходов к персонализации и рекомендательным системам в туризме; спроектирована многоуровневая архитектура (Clean Architecture и MVVM, SwiftUI, Core ML, облачные языковые модели) и реализован прототип – мобильное приложение на платформе iOS «TripSight».

Результаты. Предложена архитектура приложения и гибридная схема AI-инференса, распределяющая обработку между локальными моделями Core ML и облачными языковыми моделями по критериям задержки, приватности, доступности сети и сложности запроса. Сформулированы научная новизна и авторский вклад, рекомендации по технологическому стеку и UX-паттернам, а также определены ограничения и область применимости решения.

Ключевые слова: мобильные туристические приложения, iOS-разработка, искусственный интеллект, персонализация, рекомендательные системы, Core ML, SwiftUI, маршрутизация, UX/UI-дизайн, гибридный AI-инференс

Поступила 11.03.2026, одобрена после рецензирования 26.06.2026, принята к публикации 07.08.2026

Для цитирования. Бережной К. А. Проектирование iOS-приложения интеллектуального туристического гида с интеграцией AI-сервисов: архитектура, UX и технологии // Известия Кабардино-Балкарского научного центра РАН. 2026. Т. 28. № 4. С. 46–61. DOI: 10.35330/1991-6639-2026-28-4-46-61



Designing an iOS intelligent tourist guide application with integrated AI services: architecture, UX, and technologies

K.A. Berezhnoi

Plekhanov Russian University of Economics
36, Stremyanny lane, Moscow, 115054, Russia

Abstract. Mobile applications have become the primary channel for supporting travelers throughout all stages of their journey, while the advancement of artificial intelligence shapes user expectations for a tailored “smart guide”. However, existing travel services rarely combine deep personalization, context-aware routing, data privacy protection, and robust performance under unstable internet connection conditions.

Aim. The study aims to develop and justify an approach to designing an iOS intelligent tourist guide application with integrated AI services, including the choice of architecture, technology stack, personalization methods, and UX solutions.

Materials and methods. An analysis of 47 “Travel” applications in the App Store and 17 specialized AI services was performed; a comparative analysis of approaches to personalization and recommendation systems in tourism was conducted; a multi-tiered architecture (Clean Architecture, MVVM, SwiftUI, Core ML, cloud-based language models) was designed, and a prototype, the iOS mobile application “TripSight”, was implemented.

Results. An application architecture and a hybrid AI inference scheme have been proposed, distributing processing between local Core ML models and cloud-based language models based on criteria of latency, privacy, network availability, and request complexity. The scientific novelty and the author's contribution have been formulated, along with recommendations for the technology stack and UX patterns; furthermore, the limitations and the scope of applicability of the proposed solution have been defined.

Keywords: mobile travel applications, iOS development, artificial intelligence, personalization, recommender systems, Core ML, SwiftUI, routing, UX/UI design, hybrid AI inference

Submitted 11.03.2026,

approved after reviewing 26.06.2026,

accepted for publication 07.08.2026

For citation. Berezhnoi K.A. Designing an iOS intelligent tourist guide application with integrated AI services: architecture, UX, and technologies. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2026. Vol. 28. No. 4. Pp. 46–61. DOI: 10.35330/1991-6639-2026-28-4-46-61

ВВЕДЕНИЕ

Мобильные приложения в последние годы стали важнейшим каналом взаимодействия пользователей с цифровыми сервисами. В туристической сфере они выполняют роль помощника на всех этапах планирования и сопровождения поездки – от поиска и бронирования отелей и билетов до получения рекомендаций по достопримечательностям, навигации и анализа внешней среды. Развитие искусственного интеллекта позволило интегрировать в мобильные приложения функции интеллектуального гида: персонализированные маршруты, диалоговые консультации, анализ отзывов и мультимодальные сценарии взаимодействия [1]. Параллельно растёт качество UI/UX мобильных приложений: внедряются более

© Berezhnoi K.A., 2026



Content is available under license [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/)

понятные пользовательские сценарии, упрощается навигация, повышается скорость работы, что формирует у пользователей повышенные ожидания к туристическим сервисам¹.

Цель настоящей статьи – разработать подход к созданию мобильного приложения на платформе iOS с функционалом интеллектуального туристического гида, интегрирующего сервисы искусственного интеллекта для построения персонализированных маршрутов, а также обосновать выбор технологического стека, архитектуры, методов персонализации и UX-решений, опираясь на практику разработки мобильных приложений на iOS.

Задачи исследования:

- выполнить анализ рынка и собрать пользовательские требования к современным iOS-приложениям туристического сегмента;
- исследовать масштабы применения искусственного интеллекта в туризме и провести сравнительный анализ существующих подходов к персонализации маршрутов и рекомендаций;
- рассмотреть современные архитектурные подходы к интеграции внешних AI-API в iOS-приложения;
- определить источники данных и контекстные факторы, необходимые для работы интеллектуального помощника;
- определить требования к дизайну приложения и сформировать метрики оценки качества пользовательского опыта и рекомендаций;
- спроектировать архитектуру iOS-приложения интеллектуального гида, описать ключевые структуры и их взаимодействие;
- сформулировать набор рекомендаций и лучших практик по выбору технологического стека, архитектуре и UI/UX-организации приложений для платформы iOS.

Научная новизна и авторский вклад

Сами по себе архитектурные принципы (Clean Architecture, MVVM) и технологии (Core ML, облачные LLM) являются известными; научный вклад работы состоит не в их изобретении, а в способе их интеграции и адаптации к ограничениям мобильного туристического сценария. Авторский вклад заключается в следующем:

- Предложен способ интеграции контекстных факторов (локация, погода, загруженность, время) в рекомендационный конвейер мобильного гида, в котором контекст применяется на этапе on-device-ранжирования кандидатов, что снижает задержку и зависимость от сети по сравнению с чисто облачными рекомендателями.
- Разработана схема балансировки AI-инференса между устройством и облаком – «диспетчер инференса», распределяющий запросы по критериям задержки, приватности, доступности сети и сложности запроса; это позволяет сочетать отзывчивость и приватность on-device-обработки с качеством облачных языковых моделей.
- Предложена комбинированная схема персонализации, объединяющая начальную сегментацию, профилирование интересов, контекстно-зависимое ранжирование и оптимизацию маршрутов с ограничениями, адаптированная к мобильным ограничениям (офлайн-режим, аппаратные ресурсы, стоимость облачных запросов).
- Сформулирован обоснованный набор рекомендаций по технологическому стеку и UX-паттернам для интеллектуальных туристических ассистентов на iOS, пригодный для практического переноса на другие города и сегменты туризма.

¹AI in Tourism Marketing: Hyper-Personalization and More Bookings. Mize. URL: <https://mize.tech/blog/ai-in-tourism-marketing-hyper-personalization-and-more-bookings> (дата обращения: 10.02.2026).

В отличие от существующих обзоров и систем [2, 3], в которых рассматриваются преимущественно алгоритмы рекомендаций или серверные архитектуры, и от LLM-ориентированных подходов [1, 4] настоящая работа фокусируется на целостном проектировании клиентской части на iOS с явным разделением обязанностей между устройством и облаком.

МЕТОДИКА ИССЛЕДОВАНИЯ

1. Анализ предметной области и постановка задачи

В рамках анализа рынка мобильных туристических приложений для платформы iOS было рассмотрено 47 приложений категории «Travel» в App Store, отобранных по критериям: рейтинг не ниже 4.6, не менее 100 отзывов и релевантность функциональности (наличие поисковых и навигационных сценариев). Для каждого приложения фиксировались наличие AI-функций (чат-бот, персональные рекомендации, динамическая маршрутизация), поддержка офлайн-режима, используемые UX-паттерны и способы монетизации. Дополнительно проанализированы 17 специализированных сервисов с заявленной интеграцией технологий искусственного интеллекта. Полученные данные позволили сформулировать требования к функционалу и UX-характеристикам, легшие в основу проектируемого приложения «TripSight».

Задача проектируемого приложения (условно – «TripSight») состоит в том, чтобы облегчить выбор и составление плана путешествия и в формате диалога с AI-ассистентом, использующим персонализированную рекомендательную систему и контекстные факторы (локация, погода, загруженность объектов, время суток), построить маршрут поездки при сохранении высокого уровня приватности данных и возможности работы в условиях нестабильного интернет-соединения.

2. Источники и структура данных

Для реализации интеллектуального туристического гида необходима интеграция нескольких классов данных:

- официальные и коммерческие каталоги достопримечательностей (категории, описания, координаты, расписание, тарифы) и туристической инфраструктуры (музеи, парки, рестораны, транспорт);
- геоданные и картографические сервисы (карты, маршруты, оценка времени в пути), обеспечиваемые, например, MapKit и сторонними API;
- данные о пользователе: настройки, интересы, история просмотренных и посещенных мест, сохраненные маршруты, отзывы и оценки;
- контекстные данные в реальном времени: погода, загруженность локаций, локальные мероприятия, а также телеметрия использования приложения для последующей аналитики.

Структура данных предполагает использование локальной базы (например, Core Data или SQLite) для кеширования ключевой информации и обеспечения офлайн-режима, а также серверной части, агрегирующей внешние источники и выполняющей тяжелые аналитические запросы. Для AI-компоненты используются как облачные языковые модели, доступные по REST API, так и локальные модели, конвертированные в формат Core ML для выполнения на устройстве².

²Core ML. Apple Developer Documentation. URL: <https://developer.apple.com/documentation/coreml> (дата обращения: 05.02.2026).

3. Обоснование выбора методов персонализации и маршрутизации

Персонализация рекомендаций в туристическом приложении может базироваться на комбинации контентно-ориентированных методов, коллаборативной фильтрации и кластеризации пользователей по поведенческим паттернам. Исследования по AI-персонализации в туризме подчеркивают эффективность подходов, объединяющих профилирование интересов, анализ контекста (локация, время, погода) и предсказание вероятности отклика на конкретное предложение [1, 2].

В контексте «TripSight» целесообразно использовать следующую схему:

- начальная сегментация пользователей по типам (любители музеев, поклонники природы, гостротуристы и т. п.) на основе анкеты и первых действий в приложении;
- построение профиля интересов по истории взаимодействий и отзывам;
- применение рекомендательной модели для ранжирования достопримечательностей с учетом контекста (время, погода, близость к текущему местоположению);
- использование алгоритмов оптимизации маршрутов (например, модификации задачи коммивояжера с ограничениями по времени и предпочтениям) для построения дневных и многодневных маршрутов [2].

Ряд работ показывает, что внедрение AI-рекомендателей и динамической маршрутизации повышает вовлеченность и глубину использования туристических приложений за счет ощущения «умного сопровождения» путешественника [2]. В предлагаемой архитектуре вычислительно сложные части (кластеризация, обучение моделей) выполняются на сервере, а ранжирование кандидатов и адаптация в реальном времени – частично на устройстве средствами Core ML.

3.1. Сравнительный анализ существующих подходов к персонализации в туризме

Систематизация существующих исследований позволяет выделить пять основных классов методов персонализации в туристических рекомендательных системах и критически оценить их применимость к мобильной платформе. Контентно-ориентированная фильтрация опирается на соответствие характеристик объектов профилю интересов пользователя; она устойчива к проблеме «холодного старта» по объектам, но склонна к избыточной специализации рекомендаций. Коллаборативная фильтрация использует поведение похожих пользователей и обеспечивает обнаружение неочевидных предложений, однако чувствительна к разреженности данных и «холодному старту» по пользователям [2]. Гибридные и онтологические системы, рассмотренные в обзоре [3], сочетают оба подхода и формализуют предметную область, повышая качество и объяснимость рекомендаций ценой усложнения и трудоемкости поддержки базы знаний.

Контекстно-зависимые методы дополнительно учитывают локацию, время, погоду и социальный контекст, что критично именно для мобильного туризма [2], однако требуют надежного потока контекстных данных. Наиболее новым направлением являются подходы на основе больших языковых моделей (LLM): они обеспечивают диалоговое уточнение предпочтений и генерацию развернутых, объяснимых рекомендаций [1, 4], но остаются ресурсоемкими, зависят от сети и подвержены риску галлюцинаций. Сопоставление подходов (табл. 1) показывает, что ни один метод не покрывает все требования мобильного интеллектуального гида, поэтому в «TripSight» обоснованно применяется гибридная схема, объединяющая контекстно-зависимое ранжирование на устройстве с диалоговым LLM-слоем.

Таким образом, проведенный анализ подтверждает, что ни один класс методов не покрывает все требования мобильного интеллектуального гида. Контентная фильтрация обеспечивает стабильность при ограниченных данных, коллаборативная фильтрация

расширяет охват рекомендаций при накоплении аудитории, а LLM обеспечивают гибкий диалог за счет высоких расходов и зависимости от сети. Именно поэтому в «TripSight» обосновано применение гибридной схемы, объединяющей преимущества каждого подхода и компенсирующей их ограничения [7, 8].

Таблица 1. Сравнительный анализ подходов к персонализации в туризме

Table 1. Comparative analysis of personalization approaches in tourism

Подход	Принцип	Сильные стороны	Ограничения	Роль в «TripSight»
Контентная фильтрация	Сопоставление свойств объектов профилю	Нет «холодного старта» по объектам, прозрачность	Избыточная специализация рекомендаций	Базовое профилирование интересов
Коллаборативная фильтрация	Поведение похожих пользователей	Обнаружение неочевидных предложений	Разреженность данных, «холодный старт»	Дополнение при накоплении данных
Гибридные / онтологические	Комбинация методов и знаний домена	Качество и объяснимость	Сложность и поддержка базы знаний	Концептуальная основа ранжирования
Контекстно-зависимые	Учет локации, времени, погоды	Релевантность «здесь и сейчас»	Нужен поток контекстных данных	On-device ранжирование в реальном времени
На основе LLM	Диалог и генерация на больших моделях	Объяснимость, гибкость диалога	Ресурсоемкость, сеть, галлюцинации	Облачный диалоговый слой

4. Определение метрик качества пользовательского опыта и рекомендаций

Оценка эффективности интеллектуального туристического гида требует сочетания количественных метрик качества рекомендаций и показателей UX. Для рекомендательной части применимы традиционные метрики информационного поиска (precision, recall, NDCG) на основе откликов пользователей, а также бизнес-метрики — конверсия из просмотра в посещение, частота использования рекомендаций, средняя длина сессии.

С точки зрения UX ключевыми показателями являются:

- оценка удобства интерфейса по опросным шкалам (например, SUS);
- время выполнения типичных сценариев (поиск маршрута, добавление места, получение подсказки);
- доля пользователей, использующих офлайн-режим и AI-чат;
- показатели удержания (retention) и возвращаемости.

Эмпирические исследования UX мобильных туристических приложений показывают, что для этого сегмента критически важны простота навигации, минимизация когнитивной нагрузки и работоспособность при плохом соединении, в условиях стресса и спешки, характерных для путешествий [5]³. Дефицит доверия и удобства, выявленный у пользователей поколения Z [5], дополнительно обосновывает приоритет прозрачности и отзывчивости интерфейса. Эти положения легли в основу требований к интерфейсу «TripSight»⁴.

³How to Design a Travel App UI/UX & Case Study. Fuselab Creative. URL: <https://fuselabcreative.com/how-to-design-a-travel-app-ui-ux> (дата обращения: 10.02.2026).

⁴Hospitality and Tourism Mobile App Development: Designing Intuitive Tourism Apps. FasterCapital. URL: <https://fastercapital.com/content/Hospitality-and-tourism-mobile-app-development--User-Experience-Matters--Designing-Intuitive-Tourism-Apps.html> (дата обращения: 14.02.2026).

5. Анализ инструментов и технологий разработки iOS-приложений

Экосистема iOS предоставляет разработчикам широкий набор инструментов и библиотек для создания современных приложений с адаптивными интерфейсами и интеграции AI-инструментов различной сложности: Swift, SwiftUI, UIKit, Core ML, Vision, Core Location, MapKit и др. Декларативный фреймворк SwiftUI упрощает создание плавных и понятных интерфейсов и ускоряет прототипирование сложных экранов.

Интеграция AI-инструментов в iOS-приложения обычно выполняется двумя основными способами: через облачные API крупных провайдеров (языковые модели, внешние рекомендательные сервисы) и через локальные модели, развертываемые средствами Core ML. Эмпирические данные по on-device AI показывают, что размещение моделей на устройстве снижает задержку, уменьшает зависимость от сети и повышает приватность за счет отсутствия передачи сырых данных на сервер [6]⁵. В то же время для сложных языковых моделей и мультимодальной обработки (текст и изображение) по-прежнему целесообразны облачные сервисы, комбинируемые с локальными компонентами в гибридной архитектуре⁶.

Таблица 2. Сравнительные характеристики каналов AI-инференса

Table 2. Comparative characteristics of AI inference channels

Критерий	On-device (Core ML)	Облачные LLM (REST)
Задержка	Низкая (десяtkи–сотни мс)	Выше ($\approx 1\text{--}2$ с, зависит от сети)
Приватность	Высокая (данные на устройстве)	Ниже (передача на сервер)
Класс задач	Классификация, ранжирование	Диалог, генерация, мультимодальность
Работа офлайн	Поддерживается	Не поддерживается
Требования к устройству	Память и вычислительные ресурсы	Минимальные
Стоимость эксплуатации	Разовая (модель в приложении)	Растет с числом запросов
Качество на сложных запросах	Ограниченное	Высокое

Сравнение каналов инференса показывает, что on-device-обработка и облачные LLM являются взаимодополняющими, а не конкурирующими подходами. Локальные модели оптимальны для классификации и ранжирования с низкой задержкой и высокой приватностью, тогда как облачные модели незаменимы для сложных диалоговых сценариев и мультимодальной обработки. Диспетчер инференса динамически выбирает канал обработки по критериям таблицы 2, что позволяет реализовывать баланс между производительностью, приватностью и качеством рекомендаций [6, 9].

6. Архитектура мобильного приложения и требования

После анализа большого числа инструментов была предложена многоуровневая архитектура приложения «TripSight», которая сочетает в себе принципы Clean Architecture и MVVM для разделения слоев доменной логики, данных и представления (рис. 1). Верхний

⁵Deploy machine learning and AI models on-device with Core ML. WWDC24, Apple Developer. URL: <https://developer.apple.com/videos/play/wwdc2024/10161> (дата обращения: 12.02.2026).

⁶Machine Learning & AI. WWDC24, Apple Developer. URL: <https://developer.apple.com/machine-learning> (дата обращения: 09.02.2026).

уровень образуют SwiftUI-экраны (чат с AI-ассистентом, карта маршрута, каталог мест, профиль пользователя), взаимодействующие с ViewModel-слоем через привязки состояния. ViewModel инкапсулирует бизнес-логику: построение и изменение маршрутов, подготовку запросов к AI-сервисам, обработку ответов и управление кешем.

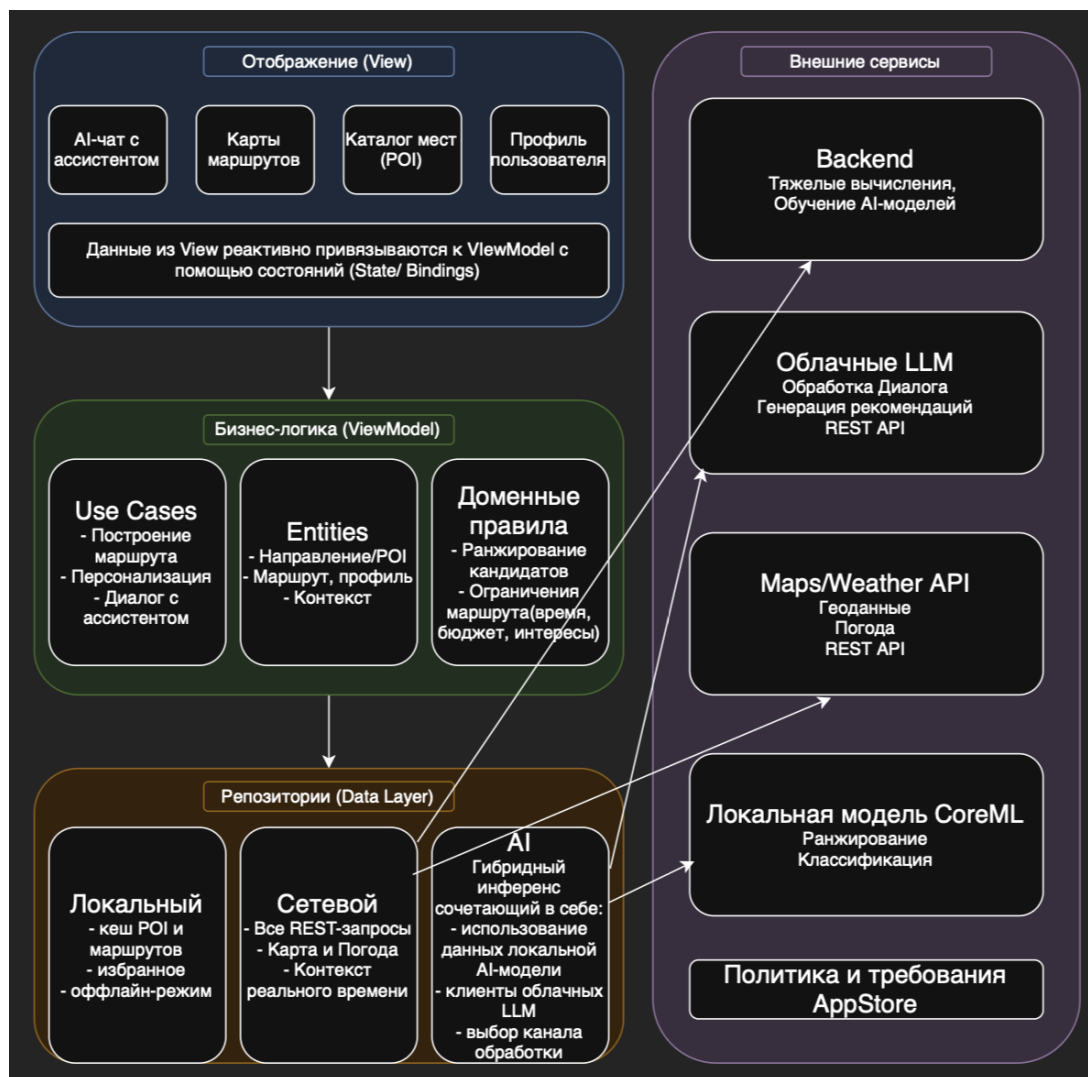


Рис. 1. Глобальная архитектура приложения «TripSight» (Clean Architecture + MVVM)

Fig. 1. Global architecture of the TripSight application (Clean Architecture + MVVM)

Архитектура, изображенная на рисунке, показывает схему взаимодействия всех слоев приложения. SwiftUI-экраны взаимодействуют только с ViewModel, не имея прямого доступа к данным, а доменная логика остается независимой от конкретных реализаций репозитория.

На уровне данных выделяются репозитории для:

- локальных данных (избранное, история, кеш POI и маршрутов);
- сетевого слоя (REST-клиенты к бэкенду, картам, погодным и AI-сервисам);
- AI-инференса (обертки над Core ML-моделями и клиентами облачных LLM).

Распределение AI-инференса между локальными и облачными компонентами иллюстрирует рисунок 2: диспетчер инференса выбирает канал обработки в зависимости от задержки, приватности, доступности сети и сложности запроса.

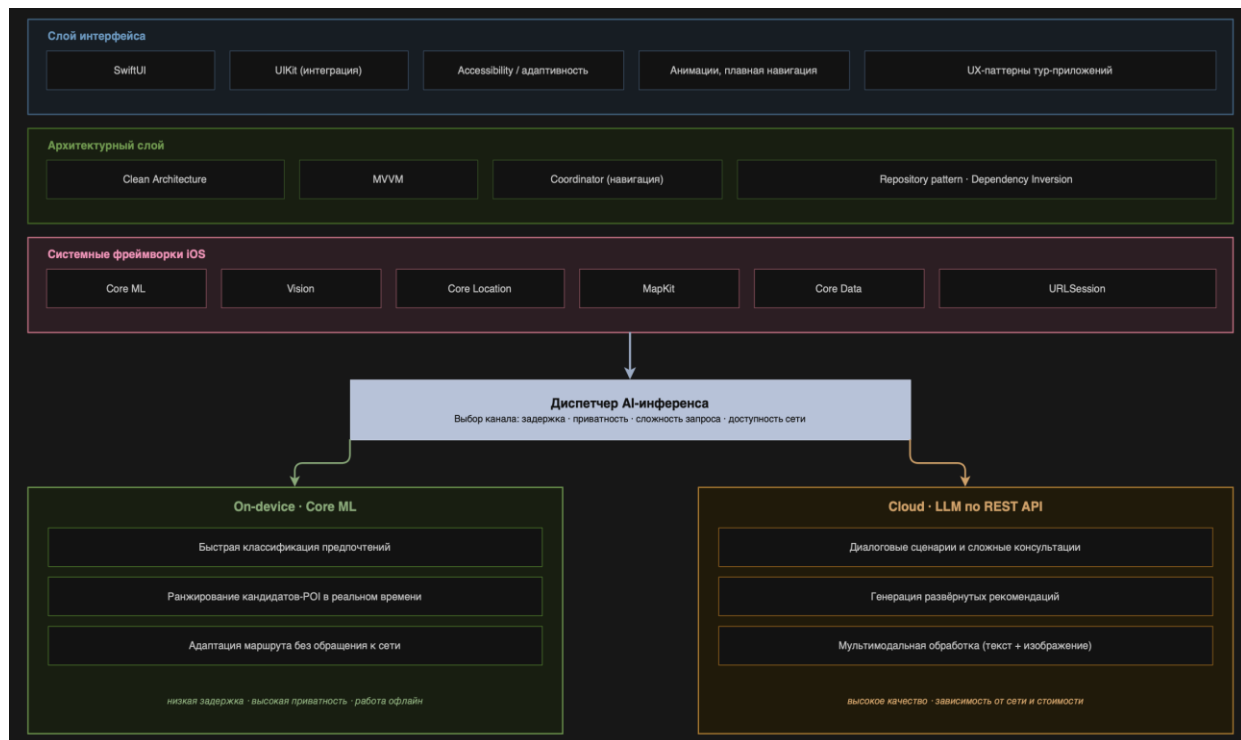


Рис. 2. Технологический стек и гибридная схема AI-инференса

Fig. 2. Tech stack and hybrid AI inference scheme

На этой схеме видно, что диспетчер инференса выступает компонентом, определяющим баланс между производительностью, приватностью и качеством ответа. Запросы с требованиями к задержке или приватности направляются в on-device-канал через Core ML, тогда как сложные диалоговые и мультимодальные задачи – в облачный LLM. Такое разделение позволяет приложению оставаться отзывчивым и работоспособным даже при нестабильном соединении, что критично в условиях туристического сценария.

Таблица 3. Состав модулей прототипа «TripSight»

Table 3. Module composition of the 'TripSight' prototype

Модуль	Назначение	Слой	Технологии
Чат с AI-ассистентом	Диалог, ввод запросов, вывод рекомендаций	Presentation	SwiftUI
Карта маршрута	Визуализация маршрута и точек интереса	Presentation	SwiftUI, MapKit
Каталог достопримечательностей	Просмотр и фильтрация POI	Presentation	SwiftUI
Профиль и настройки	Интересы, предпочтения, история	Presentation	SwiftUI
История и сохраненные маршруты	Хранение и повторный доступ	Data	Core Data / SQLite
Координатор навигации	Переходы, развязка UI и логики	Presentation	Coordinator
Сетевой слой	Запросы к backend, картам, погоде	Data	URLSession, REST
Модуль AI-инференса	Гибридная обработка запросов	Data / Domain	Core ML, REST (LLM)

Все модули приложения на платформе iOS реализуют принцип разделения ответственности: каждый модуль закреплен за конкретным слоем архитектуры и решает строго определенную задачу. Presentation-слой отвечает за отображение и пользовательские сценарии, Data-слой – за хранение и доступ к данным, а специализированный модуль AI-инференса инкапсулирует всю логику взаимодействия с интеллектуальными компонентами. Такая структура упрощает тестирование, поддержку и итеративное развитие приложения.

Таблица 4. Взаимодействие компонентов по слоям

Table 4. Interaction of components by architectural layers

Компонент	Получает	Передает	Взаимодействует с
View (SwiftUI)	Состояние от ViewModel	Пользовательские события	ViewModel
ViewModel	Результаты Use Cases	Состояние для View	Use Cases, Coordinator
Use Case (Domain)	Данные репозитория	Доменный результат	Repositories, Entities
Local Repository	Запросы домена	Кешированные данные	Core Data / SQLite
Network Repository	Запросы домена	Данные внешних API	Backend, Maps, Weather
AI Inference Repository	Запрос на инференс	Ранжирование / ответ	Core ML, облачный LLM

По схеме видно: View получает данные от ViewModel, ViewModel делегирует задачи Use Case, а Use Case обращается к репозиториям. Ни один из слоев не знает о деталях реализации вышестоящего.

Нефункциональные требования включают:

- стандартизированное время ответа AI-ассистента для диалогового режима (десятки – первые сотни миллисекунд при on-device-инференсе и 1–2 секунды при обращении к облаку);
- устойчивость при потере сети и поддержку офлайн-режима для основных сценариев (просмотр сохраненных маршрутов и описаний мест);
- плавность и адаптивность интерфейса под различные устройства экосистемы iOS, а также соблюдение требований доступности (accessibility).

7. Реализация прототипа приложения «TripSight»

В рамках исследования разработан прототип iOS-приложения «TripSight», реализующий ключевые функции интеллектуального туристического гида: формирование персонализированных маршрутов, диалоговое взаимодействие с AI-ассистентом, визуализацию маршрутов на карте и сохранение пользовательских сценариев. Прототип служит практической реализацией предложенной архитектуры, ранее описанной на концептуальном уровне, и тем самым сокращает разрыв между заявленной целью исследования и представленными результатами.

Приложение реализовано на языке Swift с использованием SwiftUI для построения интерфейса и сочетания Clean Architecture и MVVM для организации логики. Такое разделение изолирует слой представления, доменную логику и доступ к данным, упрощает масштабирование приложения и замену отдельных AI-компонентов без переработки

всей системы. Состав модулей прототипа и их распределение по слоям приведены в таблице 3, а характер взаимодействия компонентов – в таблице 4; навигация между экранами организована через отдельный координирующий слой, что снижает связность UI-компонентов и бизнес-логики.

На уровне данных в прототипе выделены три ключевых класса репозитория: локальное хранилище пользовательских данных и кеша, сетевой слой для работы с внешними API и специализированный модуль AI-инференса. Последний инкапсулирует взаимодействие как с облачными языковыми моделями через REST-интерфейсы, так и с локальными моделями в формате Core ML, что реализует гибридный подход к интеллектуальной обработке запросов [6]. Практическая реализация прототипа подтверждает применимость предложенной архитектуры к задаче построения интеллектуального туристического ассистента на платформе iOS.

8. Экспериментальная оценка предложенного подхода

Для проверки работоспособности предложенной архитектуры и оценки ее практической эффективности предусмотрена экспериментальная апробация прототипа «TripSight». Поскольку работа носит характер обоснованного архитектурного подхода, на текущем этапе формализован воспроизводимый план эксперимента: определены группы метрик, условия тестирования и способ сопоставления режимов работы системы. Оценка строится на трех группах показателей – технических, пользовательских и рекомендательных (табл. 5).

Таблица 5. План экспериментальной оценки прототипа

Table 5. Experimental evaluation plan for the prototype

Категория	Метрики	Что оценивает	Способ измерения
Технические	Время первого ответа AI; полное время построения маршрута; число сетевых запросов; устойчивость при нестабильной сети	Производительность и стабильность гибридной архитектуры	Инструментирование, логирование сценариев
Пользовательские	Время выполнения сценария; число действий до результата; SUS; доля завершивших без ошибок	Удобство и эффективность взаимодействия	Юзабилити-тест (5–10 пользователей), анкета SUS
Рекомендательные	precision / recall / NDCG; доля принятых рекомендаций; число сохраненных и измененных маршрутов	Качество персонализации	Сравнение со стандартным (неперсонализированным) маршрутом

Этот план оценки охватывает аспекты работоспособности предложенной архитектуры. Технические метрики позволят оценить производительность гибридного AI-инференса, пользовательские – зафиксировать удобство и эффективность взаимодействия с приложением, а рекомендационные – оценить качество персонализации в сравнении со стандартным маршрутом. Комплексность плана обеспечивает воспроизводимость результатов и возможность независимой верификации полученных данных [10].

Для оценки качества персонализации применяется сравнительный подход: сопоставляется сценарий со стандартным маршрутом и сценарий с персонализированной рекомендацией, учитывающей интересы, контекст и предыдущие взаимодействия. По результатам апробации в итоговой редакции статьи планируется представить численные значения метрик и графический материал, иллюстрирующий различия между сравниваемыми режимами на выборке из 5–10 пользователей. Такой формат позволяет обосновать, что предложенная архитектура не только логически выстроена, но и демонстрирует измеримую эффективность.

9. Ограничения и область применимости архитектуры

Анализ ограничений и условий применимости является важным элементом научного описания архитектурного решения. Первым ограничением является зависимость части функциональности от качества интернет-соединения: несмотря на локальное кеширование и on-device-компоненты Core ML, сложные диалоговые сценарии и генерация расширенных рекомендаций через облачные языковые модели чувствительны к задержкам сети и доступности внешних API [6].

Вторым ограничением выступают аппаратные характеристики устройства: локальные модели снижают задержку и повышают приватность, но требуют вычислительных ресурсов, памяти и пространства в хранилище, что затрудняет работу на старых устройствах. Третьим ограничением является стоимость эксплуатации облачной AI-инфраструктуры: при росте числа пользователей и интенсивности запросов расходы на внешние сервисы становятся значимым фактором, влияющим на распределение вычислительной нагрузки между устройством и сервером.

Подход наиболее уместен для туристических приложений, где требуется сочетание персонализированных рекомендаций, контекстной маршрутизации, офлайн-доступа к части данных и интерактивного интерфейса, и может оказаться избыточным для простых справочных сервисов без AI-составляющей. Явное обсуждение ограничений повышает научную добросовестность работы и делает выводы более обоснованными.

НАУЧНАЯ НОВИЗНА И ВКЛАД

Сами по себе архитектурные принципы (Clean Architecture, MVVM) и технологии (Core ML, облачные LLM) являются известными; научный вклад работы состоит не в их изобретении, а в способе их интеграции и адаптации к ограничениям мобильного туристического сценария. Авторский вклад заключается в следующем:

- Предложен способ интеграции контекстных факторов (локация, погода, загруженность, время) в рекомендательный конвейер мобильного гида, в котором контекст применяется на этапе on-device-ранжирования кандидатов, что снижает задержку и зависимость от сети по сравнению с чисто облачными рекомендателями.

- Разработана схема балансировки AI-инференса между устройством и облаком – «диспетчер инференса», распределяющий запросы по критериям задержки, приватности, доступности сети и сложности запроса (рис. 2); это позволяет сочетать отзывчивость и приватность on-device-обработки с качеством облачных языковых моделей.

- Предложена комбинированная схема персонализации, объединяющая начальную сегментацию, профилирование интересов, контекстно-зависимое ранжирование и оптимизацию маршрутов с ограничениями, адаптированная к мобильным ограничениям (офлайн-режим, аппаратные ресурсы, стоимость облачных запросов).

– Сформулирован обоснованный набор рекомендаций по технологическому стеку и UX-паттернам для интеллектуальных туристических ассистентов на iOS, пригодный для практического переноса на другие города и сегменты туризма.

В отличие от существующих обзоров и систем [2, 3], в которых рассматриваются преимущественно алгоритмы рекомендаций или серверные архитектуры, и от LLM-ориентированных подходов [1, 4] настоящая работа фокусируется на целостном проектировании клиентской части на iOS с явным разделением обязанностей между устройством и облаком.

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

Результатом исследования является обоснованная концепция и проект iOS-приложения интеллектуального туристического гида, основанного на персонализированных данных и контекстных рекомендациях, а также реализованный прототип, подтверждающий техническую осуществимость подхода. Система сочетает использование внешних языковых моделей для сложных диалоговых сценариев и локальных моделей Core ML для быстрых задач классификации предпочтений и ранжирования объектов, что улучшает отклик интерфейса и снижает зависимость от сетевой инфраструктуры. Соответствие поставленных задач полученным результатам приведено в таблице 6.

Таблица 6. Соответствие задач исследования полученным результатам

Table 6. Mapping of research objectives to the results obtained

Задача исследования	Полученный результат	Раздел
Анализ рынка и сбор требований	Проанализированы 47 приложений и 17 AI-сервисов; сформулированы функциональные и UX-требования	Методика, п. 1
Анализ подходов к AI-персонализации	Выполнен сравнительный анализ пяти классов подходов с критикой и научными источниками	п. 3.1, табл. 1
Архитектурные подходы интеграции AI-API	Обоснована гибридная схема (Clean + MVVM, Core ML + облачные LLM)	п. 5–6, рис. 1–2
Источники данных и контекст	Определены классы данных и контекстные факторы	п. 2
Требования к дизайну и метрики UX	Определены SUS, время сценариев, retention, конверсия; составлен план измерений	п. 4, 8, табл. 5
Проектирование архитектуры	Спроектирована многоуровневая архитектура; описаны модули и их взаимодействие	п. 6, рис. 1, табл. 3–4
Рекомендации по стеку и UX	Сформулированы рекомендации и критерии выбора инструментов	Рекомендации

Сопоставление задач и результатов подтверждает, что все поставленные цели исследования достигнуты. Проведен комплексный анализ рынка и подходов к персонализации, спроектирована и реализована гибридная архитектура, сформулированы метрики качества и план экспериментальной апробации. Совокупность полученных результатов формирует целостную научно обоснованную основу для практической разработки интеллектуального туристического ассистента на платформе iOS.

РЕКОМЕНДАЦИИ ПО ВЫБОРУ ТЕХНОЛОГИЧЕСКОГО СТЕКА И UX-ПАТТЕРНОВ

На основании анализа литературы и практики разработки при создании интеллектуальных туристических приложений на iOS рекомендуется придерживаться следующих принципов:

- использовать SwiftUI как основной фреймворк интерфейса с возможностью интеграции UIKit для специализированных компонентов (карты, сложные кастомные контролы);
- строить архитектуру на принципах Clean Architecture и MVVM с четким разделением слоев, что упрощает масштабирование AI-функций и замену сервисов;
- комбинировать облачные AI-сервисы (LLM, мультимодальные модели) с локальными моделями Core ML для баланса между качеством, задержкой и приватностью [6];
- соблюдать принципы UX для туристических приложений: простота навигации, визуальная иерархия, офлайн-доступ, адаптивность и доступность [5];
- встраивать механизмы сбора UX-метрик и обратной связи для итеративного улучшения качества рекомендаций и интерфейса.

КРИТЕРИИ ВЫБОРА МЕТОДОВ И ИНСТРУМЕНТОВ

Ключевыми критериями выбора моделей рекомендательных систем являются устойчивость к «шуму данных», возможность дообучения и корректная работа при ограниченном объеме пользовательских данных [2, 3]. При выборе AI-инструментов для интеграции в iOS-приложения необходимо учитывать производительность на устройствах, наличие средств профилирования и оптимизации, а также соответствие требованиям магазина приложений и политике конфиденциальности. Сторонние облачные AI-сервисы целесообразно оценивать по стоимости, ограничениям использования, наличию SDK и качеству документации.

ОБСУЖДЕНИЕ И ЗАКЛЮЧЕНИЕ

Проведенное исследование показывает, что интеграция искусственного интеллекта в iOS-приложения туристического профиля позволяет перейти от статических каталогов с жестко заданными маршрутами к персонализированным, динамическим, контекстно-зависимым сценариям сопровождения путешественника. Комбинация диалогового AI-ассистента и интеллектуальной маршрутизации формирует у пользователя ощущение «цифрового гида», адаптирующегося к его стилю путешествий и внешним условиям в реальном времени [1, 2].

Предложенная архитектура приложения «TripSight» и сформулированные требования к UX/UI и технологическому стеку могут послужить основой для практической реализации интеллектуального туристического ассистента и адаптации под другие города и сегменты туризма. Перспективными направлениями дальнейшей работы являются экспериментальная оценка влияния AI-персонализации на вовлеченность пользователей и разработка методов объяснимого AI для туристических рекомендаций [4].

Благодарности. Выражаю признательность научному руководителю Китовой Ольге Викторовне, доктору экономических наук, доценту, заведующей кафедрой информатики, за содействие, помощь и консультации при проведении данного исследования.

СПИСОК ЛИТЕРАТУРЫ / REFERENCES

1. Deliyannis I., Mazi I.A., Poulimenou S.M. et al. AI-Driven personalization in tourism: Enhancing visitor experience. *Proceedings of the International Conference on Museum Big Data (MBD 2024), Athens, Greece, 18–19 November 2024. CEUR Workshop Proceedings*. Vol. 4021. Paper 14. URL: <https://ceur-ws.org/Vol-4021/paper14.pdf>
2. Gavalas D., Konstantopoulos C., Mastakas K., Pantziou G. Mobile recommender systems in tourism. *Journal of Network and Computer Applications*. 2014. Vol. 39. P. 319–333. DOI: 10.1016/j.jnca.2013.04.006
3. Borràs J., Moreno A., Valls A. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*. 2014. Vol. 41. No. 16. Pp. 7370–7389. DOI: 10.1016/j.eswa.2014.06.007
4. Nikas I.A., Koutras A., Antonopoulou T. A Conceptual framework for developing and evaluating personalized tourist recommendation systems using large language models. Innovation and creativity in tourism, business and social sciences (IACuDiT 2024). *Springer Proceedings in Business and Economics*. Cham: Springer, 2025. DOI: 10.1007/978-3-031-78471-2_20
5. Huang H., Othman A., Muhaiyuddin N.D.M. User experience deficiencies in mobile tourism applications: A preliminary study from generation Z and tourism practitioners. *International Journal of Advanced Computer Science and Applications (IJACSA)*. 2025. Vol. 16. No. 11. Pp. 261–270. DOI: 10.14569/IJACSA.2025.0161128
6. Li Y., Dang X., Tian H., Sun T., Wang Z., Ma L., Klein J., Bissyandé T.F. An empirical study of AI techniques in mobile applications. *Journal of Systems and Software*. 2024. Vol. 219. Art. 112233. DOI: 10.1016/j.jss.2024.112233
7. Кумова Д. М. Использование платформ на базе искусственного интеллекта в сфере туризма // Сервис в России и за рубежом. 2021. Т. 15. № 3. С. 18–26. DOI: 10.24412/1995-042X-2021-3-18-26
Kumova D.M. The use of artificial intelligence-based platforms in tourism. *Servis v Rossii i za rubezhom* [Services in Russia and Abroad]. Vol. 15. No. 3. Pp. 18–26. DOI: 10.24412/1995-042X-2021-3-18-26. (In Russian)
8. Ляликова В. Г., Безрядин М. М. Построение гибридной рекомендательной системы // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. 2021. № 4. С. 121–129. DOI: 10.17308/sait.2021.4/3802
Lyalikova V.G., Bezryadin M.M. Development of a hybrid recommendation system. *Proceedings of Voronezh State University. Series: Systems Analysis and Information Technologies*. 2021. No. 4. Pp. 121–129. DOI: 10.17308/sait.2021.4/3802. (In Russian)
9. Маринин А. К. Выбор архитектуры для мобильных приложений // Известия Кабардино-Балкарского научного центра РАН. 2024. Т. 26. № 5. С. 84–93. DOI: 10.35330/1991-6639-2024-26-5-84-93. EDN: LEFHZL
Marinin A.K. Choosing an architecture for mobile applications. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2024. Vol. 26. No. 5. Pp. 84–93. DOI: 10.35330/1991-6639-2024-26-5-84-93. EDN: LEFHZL. (In Russian)
10. Дергачёв К. В., Эссам М. Ф. Н. Разработка автоматизированной системы для создания и тестирования прототипов пользовательских интерфейсов // Эргодизайн. 2022. № 4. С. 252–261. DOI: 10.30987/2658-4026-2022-4-252-261
Dergachev K., Essam M.F.N. Development of an automated system for creating and testing user interface prototypes. *Ergodesign*. 2022. No. 4. Pp. 252–261. DOI: 10.30987/2658-4026-2022-4-252-261. (In Russian)

Финансирование. Исследование проведено без спонсорской поддержки.

Funding. The study was performed without external funding.

Информация об авторе

Бережной Кирилл Александрович, магистрант кафедры информатики, Российский экономический университет имени Г. В. Плеханова;

115054, Россия, Москва, Стремянный переулок, 36;

unknownkirya@gmail.com, ORCID: <https://orcid.org/0009-0000-2693-3077>

Information about the author

Kirill A. Berezhnoi, Master Student, Department of Informatics, Plekhanov Russian University of Economics;

36, Stremyanny lane, Moscow, 115054, Russia;

unknownkirya@gmail.com, ORCID: <https://orcid.org/0009-0000-2693-3077>